

### Collections

1. What is collection? Name python's most common collections with proper examples.
2. Write a python snippet that, for a list *mylist*, prints out each item and the corresponding index, one item in a line.
3. Show insertion, deletion and searching in a python dictionary with appropriate example.

### Strings

1. Write a code snippet to create a string 'ACATCCATGCATTCAT' where the letter sequence 'CAT' occurs only once.
2. Write a code snippet to create a string "blablablabla" where the letter sequence 'bla' occurs only once.
3. State the output of the following code snippet:  

```
seq = 'ACACGCTA' + 'GATTTA'
print seq[3:-1]
```
4. Write a code snippet to combine given the two strings S1 and S2 into one string.

### Functions

1. Write a function to calculate GC content of a sequence.
2. Write a function *occur(dna, pattern)* that returns the number of occurrences of a pattern in the given DNA string (dna). For example, calling the function with 'ATCCTGCTATCCATCTAT' and 'ATC' should return 3.
3. Write a function that will return the base ('A', 'C', 'G', 'T') counts.

### Control Statements

1. Define an infinite loop with appropriate example.
2. State what control statements are used in python with appropriate example.
3. Write a program to calculate the sum of only odd numbers from 1 to n using a loop.
4. Write a program to calculate the sum of even numbers from 0 to n using a loop.
5. Write a program to calculate the length of a sequence without using the function len().
6. Write a program to count the nucleotides of a sequence.
7. Write a program to make a dot plot for two sequences.
8. Write a program to generate a random nucleotide sequence of length N.
9. Write a program to generate a random protein sequence of length N.
10. Write a program to find point mutations in two given sequences S1 and S2.
11. Write a program to generate all permutations of the 3 nucleotide, i.e. codons.
12. Write a program to find the consensus sequence of given some sequences.
13. Write a program to display all the k-mers of a sequence and given k.
14. Write a program to digest a protein into peptides.
15. Write a program to align two sequences.
16. Write a program to find the open reading frame (ORF) in a sequence.
17. Write a program to find the longest ORF in a sequence.

### **Files**

1. State the modes of opening a file, with appropriate example.
2. Write a program to read sequences from a .fasta file where each sequence is not longer than one line.
3. Write a program to read sequences from a .fasta file where each sequence is not longer than one line. Calculate the GC content of each sequence and write them in another file with the corresponding sequence id of the sequences.
4. Write a program to read 100 numbers from a file input.txt, each written in a line, calculate the sum of only positive even numbers and write the output into another file output.txt.
5. Write a program to read sequences from a .fasta file where the file contains multiple sequences and each sequence is split in multiple lines.
6. Write a program to read from and write to a .csv file.
7. Write a program to parse a .xml file using biopython.
8. Write a program to display a newick tree file.

### **Biopython**

1. Write a function in python for calculating the GC content of a sequence using biopython.
2. Write a program to calculate the reverse complement of a sequence using biopython.
3. Write a program to translate a DNA sequence to a protein sequence using biopython.
4. Write a program to parse a .fasta file using biopython.
5. Write a program to parse a genbank file using biopython.
6. Write a program to reverse complement a sequence using biopython.
7. Write a program to parse a .xml file using biopython.
8. Write a program to perform a sequence alignment using biopython.
9. Write a program to read a sequence alignment using biopython.

### **Trivial**

1. Define complexity. State how the complexity of an algorithm is measured.
2. Define algorithm, the time and memory complexity of an algorithm.
3. State the best case and worst case scenario of an algorithm.
4. Name python's primitive data types.
5. Write a code snippet to measure the CPU time for a task.
6. Write a program to do a random mutation in a given DNA/protein sequence.
7. Define open reading frame (ORF).